This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

PORTAL
US Patent & Trademark Office
Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
relationship AND interface AND structure
THE ACM DIGITAL LIBRARY
Feedback Report a problem Satisfaction survey

Terms used **relationship** AND **interface** AND **structure** Found **52,559** of **140,980**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Representing extended entity-relationship structures in relational databases: a modular approach
Victor M. Markowitz, Arie Shoshani
September 1992 **ACM Transactions on Database Systems (TODS)**, Volume 17 Issue 3
Full text available: pdf(3.18 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A common approach to database design is to describe the structures and constraints of the database application in terms of a semantic data model, and then represent the resulting schema using the data model of a commercial database management system. Often, in practice, Extended Entity-Relationship (EER) schemas are translated into equivalent relational schemas. This translation involves different aspects: representing the EER schema using relational constructs, assigning n ...

Keywords: database design, extended entity-relationship model, relational data model, schema translation, semantic data model

2 DON: user interface presentation design assistant
Won Chul Kim, James D. Foley
August 1990 **Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology**
Full text available: pdf(1.33 MB) Additional Information: full citation, references, citings, index terms

3 An object-oriented approach to graphical interfaces
Paul S. Barth
April 1986 **ACM Transactions on Graphics (TOG)**, Volume 5 Issue 2
Full text available: pdf(2.23 MB) Additional Information: full citation, abstract, references, citings, index terms, review

An object-oriented system for building graphical interfaces to programs is discussed. The system, called GROW, facilitates the process of creating interfaces that are highly interactive (including direct manipulation and animation), rich in layout structure, and effectively reusable across applications. These properties are achieved through three techniques: object-based graphics with taxonomic inheritance, interobject relationships such as composition and graphical dependency, and separati ...

4 Data model for extensible support of explicit relationships in design databases
Joan Peckham, Bonnie MacKellar, Michael Doherty

April 1995 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 4 Issue 2

Full text available: pdf(2.01 MB) Additional Information: full citation, abstract, references, citings

We describe the conceptual model of SORAC, a data modeling system developed at the University of Rhode Island. SORAC supports both semantic objects and relationships, and provides a tool for modeling databases needed for complex design domains. SORAC's set of built-in semantic relationships permits the schema designer to specify enforcement rules that maintain constraints on the object and relationship types. SORAC then automatically generates C++ code to maintain the specified enforcement rules ...

Keywords: computer-aided architectural design, database constraints, relationship semantics, semantic and object-oriented data modeling

5 Towards a formal specification method for graphical user interfaces using modularized graph grammars

M. Goedicke, B. E. Sucrow

March 1996 **Proceedings of the 8th International Workshop on Software Specification and Design**

Full text available: pdf(1.31 MB) Additional Information: full citation, abstract

Publisher Site

Well designed graphical user interfaces offer a high potential to increase the productivity of human users. The necessary condition for such a good performance is that the user interface represents the semantics of the underlying application in a clear and comprehensible way. This means, especially, that not only syntactical layout but also semantic consistency conditions between the various interaction objects have to be presented in a graphical user interface appropriately. This is usually ter ...

Keywords: consistency conditions, formal specification, formal specification method, graph grammar based approach, graph grammars, graphical interaction objects, graphical man machine interfaces, graphical user interfaces, interaction objects, modularization technique, modularized graph grammars, representation schemes, semantic consistency, semantic consistency conditions, semantic feedback, semantics, syntactical layout

6 Human-computer interface development: concepts and systems for its management

H. Rex Hartson, Deborah Hix

March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1

Full text available: pdf(7.97 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Human-computer interface management, from a computer science viewpoint, focuses on the process of developing quality human-computer interfaces, including their representation, design, implementation, execution, evaluation, and maintenance. This survey presents important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is th ...

7 Towards modeling individual and collaborative construction of jigsaws using task knowledge structures (TKS)

Hilary Johnson, Joanne Hyde

December 2003 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 10 Issue 4

Full text available: pdf(410.91 KB) Additional Information: full citation, abstract, references, index terms

Recent years have seen an overwhelming interest in how people work together as a group. Both the nature of collaboration and research into how people collaborate is complex and multifaceted, with different research agendas, types of studies, and variations in the

behavioral data collected. A better understanding of collaboration is needed in order to be able to make contributions to the design of systems to support collaboration and collaborative tasks. In this article, we combine relevant liter ...

Keywords: Modeling, collaborative studies, task analysis

8 A comparative study of language support for generic programming
Ronald Garcia, Jaakko Jarvi, Andrew Lumsdaine, Jeremy Siek, Jeremiah Willcock
October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications**, Volume 38 Issue 11

Full text available: pdf(237.38 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Many modern programming languages support basic generic programming, sufficient to implement type-safe polymorphic containers. Some languages have moved beyond this basic support to a broader, more powerful interpretation of generic programming, and their extensions have proven valuable in practice. This paper reports on a comprehensive comparison of generics in six programming languages: C++, Standard ML, Haskell, Eiffel, Java (with its proposed generics extension), and Generic C. By implementi ...

Keywords: C#, C++, Eiffel, Haskell, Java, generic programming, generics, polymorphism, standard ML

9 The integrality of speech in multimodal interfaces
Michael A. Grasso, David S. Ebert, Timothy W. Finin
December 1998 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 5 Issue 4

Full text available: pdf(179.44 KB) Additional Information: full citation, abstract, references, citings, index terms

A framework of complementary behavior has been proposed which maintains that direct-manipulation and speech interfaces have reciprocal strengths and weaknesses. This suggests that user interface performance and acceptance may increase by adopting a multimodal approach that combines speech and direct manipulation. This effort examined the hypothesis that the speed, accuracy, and acceptance of multimodal speech and direct-manipulation interfaces will increase when the modalities match the per ...

Keywords: direct manipulution, input devices, integrality, medical informatics, multimodal, natural-language processing, pathology, perceptual structure, separability, speech recognition

10 Systems: Structure from anarchy: meta level representation of expert system propositions for natural language interfaces
Galina Datskovsky Moerdler
February 1988 **Proceedings of the second conference on Applied natural language processing**

Full text available: pdf(685.46 KB) Publisher Site Additional Information: full citation, abstract, references

In this paper we describe a meta level representation used for mapping natural language input into propositions of an expert system. This representation is based on verb classes that are structured hierarchically, with more general information encoded in the top level nodes and more specific information in the lower level nodes. Because of its structure, the representation is able to provide a detailed classification of the propositions, supplying a basis for defining semantics. It allows the sy ...

11 Foundations for the Arcadia environment architecture

Richard N. Taylor, Frank C. Belz, Lori A. Clarke, Leon Osterweil, Richard W. Selby, Jack C. Wileden, Alexander L. Wolf, Michael Young

November 1988 **Proceedings of the third ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments**, Volume 13 , 24 Issue 5 , 2

Full text available: pdf(2.01 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Early software environments have supported a narrow range of activities (programming environments) or else been restricted to a single "hard-wired" software development process. The Arcadia research project is investigating the construction of software environments that are tightly integrated, yet flexible and extensible enough to support experimentation with alternative software processes and tools. This has led us to view an environment as being composed of tw ...

12 DBMS implementation experience: An entity-based database user interface

R. G. G. Cattell

May 1980 **Proceedings of the 1980 ACM SIGMOD international conference on Management of data**

Full text available: pdf(815.00 KB) Additional Information: full citation, abstract, references, citings

A user interface to a database designed for casual, interactive use is presented. The interface is *entity-based:* the data display to the user is based upon entities (e.g., persons, documents, organizations) that participate in relationships, rather than upon relations alone as in the relational data model. Examples from an implementation of the system are shown, for a prototype personal database (PDB), developed in connection with the ZOG system at Carnegie-Mellon University (Robertson et ...

13 Lessons from developing audio HTML interface

Frankie James

January 1998 **Proceedings of the third international ACM conference on Assistive technologies**

Full text available: txt(50.18 KB) Additional Information: full citation, references, citings, index terms

Keywords: HTML, WWW, audio interfaces, blind, human-computer interaction

14 Trading and negotiating stream bindings

H. O. Rafaelsen, F. Eliassen

April 2000 **IFIP/ACM International Conference on Distributed systems platforms**

Full text available: pdf(161.26 KB) Additional Information: full citation, abstract, references, citings

Distributed multimedia information systems require a range of different interaction styles ranging from simple remote operation interaction to complex patterns of interaction involving both discrete and continuous data. The standardized reference model for Open Distributed Processing (ODP) defines a binding model that encapsulates different interaction styles within explicit binding objects. In this paper we discuss mechanisms for selecting and negotiating appropriate explicit stream bindings ...

15 Study of a generic graphics package for the development of various user interfaces

Amir A. Khwaja, Raghu Mannam, Joseph E. Urban

April 1992 **Proceedings of the 1992 ACM/SIGAPP Symposium on Applied computing: technological challenges of the 1990's**

Full text available: pdf(570.50 KB) Additional Information: full citation, references, index terms

16 Controlling access in multiuser interfaces

Prasun Dewan, Honghai Shen

March 1998 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 5 Issue 1

Full text available: pdf(182.07 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Traditionally, access control has been studied in the areas of operating systems and database management systems. With the advent of multiuser interfaces, there is a need to provide access control in the user interface. We have developed a general framework for supporting access control in multiuser interfaces. It is based on the classical notion of an access matrix, a generalized editing-based model of user-application interaction, and a flexible model of user-user coupling. It has been de ...

Keywords: access control, collaboration, computer-supported cooperative work, groupware, privacy, security, structure editors, user interface management systems

17 On the automation of code generation for user interface models

L. C. M. Nova, D. D. Cowan, A. v. Staa, C. J. P. Lucena

October 1994 **Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(143.52 KB) Additional Information: full citation, abstract, references, index terms

User interfaces for highly interactive software systems are usually handcrafted in spite of the availability of tools for automatic code generation. Since user interfaces typically constitute 50 to 80% of an interactive system, such automation should provide significant increases in productivity and software quality. We outline in this paper a design approach to user interfaces that will support partially automatic code generation.The design approach is based on the Abstract Data View (ADV) whic ...

18 A research topology for object-oriented analysis and design

David E. Monarchi, Gretchen I. Puhr

September 1992 **Communications of the ACM**, Volume 35 Issue 9

Full text available: pdf(9.33 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: analysis, modeling

19 7a—Capturing Meaning: Open hypermedia as a navigational interface to ontological information spaces

Mark J. Weal, Gareth V. Hughes, David E. Millard, Luc Moreau

September 2001 **Proceedings of the twelfth ACM conference on Hypertext and Hypermedia**

Full text available: pdf(275.20 KB) Additional Information: full citation, abstract, references, index terms

Ontologies provide a powerful tool for distributed agent-based information systems. However, in their raw form they can be difficult for users to interact with directly. Different query architectures use structured query languages as an interface but these still require the users to have an expert understanding of the underlying ontologies.

By using an Open Hypermedia model as an interface to an ontological information space, users can interact with such a system using familiar browsi ...

Keywords: Agent Based Systems, Fundamental Open Hypermedia Model (FOHM), Ontological Information Spaces

20 Locus looks at the Turing play: hypertextuality vs. full programmability
Jim Rosenberg
May 1998 **Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space---structure in hypermedia systems: links, objects, time and space---structure in hypermedia systems**
Full text available: pdf(1.05 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore RELEASE 1.8

Welcome
United States Patent and Trademark Office

IEEE Xplo
1 Million Doc
1 Million User

» Search Res

Help FAQ Terms IEEE Peer Review

Quick Links

Welcome to IEEE Xplore

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

IEEE Enterprise

- Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **198** of **1060766** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

relationship<and>interface<and>structure Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Complex and composite objects in CAD/CAM databases**
Wilkes, W.; Klahold, P.; Schlageter, G.;
Data Engineering, 1989. Proceedings. Fifth International Conference on , 6-10 Feb. 1989
Pages:443 - 450

[Abstract] [PDF Full-Text (632 KB)] **IEEE CNF**

---

2 **Service provisioning data relationship model for ISDN network maintenance**
King, N.J.;
Selected Areas in Communications, IEEE Journal on , Volume: 6 , Issue: 4 , May 1988
Pages:727 - 731

[Abstract] [PDF Full-Text (480 KB)] **IEEE JNL**

---

3 **Multi-functional visualization system for structure analysis**
Tao, H.; Tadamura, K.; Nakamae, E.;
Computer Graphics and Applications, 1997. Proceedings., The Fifth Pacific Conference on , 13-16 Oct. 1997
Pages:198 - 205, 223

[Abstract] [PDF Full-Text (1804 KB)] **IEEE CNF**

---

4 **Visual representation of database queries using structural similarity**
Groth, D.P.;
Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on , 16-18 July 2003
Pages:102 - 107

[Abstract] [PDF Full-Text (457 KB)] **IEEE CNF**

---

5 **Augmenting user interfaces for digital libraries with virtual reality**
Chaomei Chen;
System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on , Volume: 2 , 6-9 Jan. 1998
Pages:148 - 157 vol.2

[Abstract] [PDF Full-Text (1080 KB)] **IEEE CNF**

---

6 **InfoCrystal: A visual tool for information retrieval**
Spoerri, A.;
Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on , 25-29 Oct. 1993
Pages:150 - 157

[Abstract] [PDF Full-Text (656 KB)] **IEEE CNF**

---

7 **EDIF procedural interface to CAD data**
Lay, P.;
Electronic Interchange Format - EDIF, IEE Colloquium on , 16 Nov 1988
Pages:4/1 - 4/5

[Abstract] [PDF Full-Text (156 KB)] **IEE CNF**

---

8 **Role-based exploration of object-oriented programs**
Demsky, B.; Rinard, M.;
Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on , 19-25 May 2002
Pages:313 - 324

[Abstract] [PDF Full-Text (1223 KB)] **IEEE CNF**

---

9 **Test structures for analyzing proton radiation effects in bipolar technologies**
Barnaby, H.J.; Schrimpf, R.D.; Galloway, K.F.; Ball, D.R.; Pease, R.L.; Fouillat, P.;
Semiconductor Manufacturing, IEEE Transactions on , Volume: 16 , Issue: 2 , May 2003
Pages:253 - 258

[Abstract] [PDF Full-Text (955 KB)] **IEEE JNL**

---

10 **3D-List: a data structure for efficient video query processing**
Chih-Chih Liu; Chen, A.L.P.;
Knowledge and Data Engineering, IEEE Transactions on , Volume: 14 , Issue: 1 , Jan.-Feb. 2002
Pages:106 - 122

[Abstract] [PDF Full-Text (759 KB)] **IEEE JNL**

---

11 **Structure and magnetic properties of (001) NiFe/NiMn/Co**
Chih-Huang Lai; Yung-Hung Wang; Lien, W.C.; Lo, C.K.;
Magnetics, IEEE Transactions on , Volume: 36 , Issue: 5 , Sept 2000
Pages:2641 - 2643

[Abstract] [PDF Full-Text (72 KB)] **IEEE JNL**

---

12 **Isotype heterojunctions with flat valence or conduction band**

Babic, D.I.; Dohler, G.H.; Bowers, J.E.; Hu, E.L.;
Quantum Electronics, IEEE Journal of , Volume: 33 , Issue: 12 , Dec. 1997
Pages:2195 - 2198

[Abstract] [PDF Full-Text (96 KB)] **IEEE JNL**

---

13 **Implementing a relational database for an accelerated-life-test facility**
Barton, R.R.;
Reliability, IEEE Transactions on , Volume: 43 , Issue: 1 , March 1994
Pages:11 - 21

[Abstract] [PDF Full-Text (964 KB)] **IEEE JNL**

---

14 **A case history analysis of software error cause-effect relationships**
Nakajo, T.; Kume, H.;
Software Engineering, IEEE Transactions on , Volume: 17 , Issue: 8 , Aug. 1991
Pages:830 - 838

[Abstract] [PDF Full-Text (720 KB)] **IEEE JNL**

---

15 **Dielectric breakdown strength affected by the lamellar configuration in XLPE insulation at a semiconducting interface**
Okamoto, T.; Ishida, M.; Hozumi, N.;
Electrical Insulation, IEEE Transactions on [see also Dielectrics and Electrical Insulation, IEEE Transactions on] , Volume: 24 , Issue: 4 , Aug. 1989
Pages:599 - 607

[Abstract] [PDF Full-Text (644 KB)] **IEEE JNL**

---

1 2 3 4 5 6 7 8 9 10 11 12 13 14 Next

Copyright © 2004 IEEE — All rights reserved

h eee e eee g e ch e ch e e e h c e e

Google Web Images Groups News Froogle more »
relationship interface structure software prd Search Advanced Search Preferences

Web Results **1** - **10** of about **23,700** for **relationship interface structure software program hash table**. (**0.32** seconds

REALbasic - encyclopedia article about REALbasic. Free access, no ...
... features of the traditional Macintosh user **interface** on top ... ex' , both to emphasize the **relationship** with Unix ... is an information set with a regular **structure**. ...
encyclopedia.thefreedictionary.com/REALbasic - 31k - Cached - Similar pages

Software Construction and Data **Structures** with Ada 95, 2nd Edition
... Sometimes only a package **interface** is given, so the ... A discussion of the **relationship** between performance predictionand ... is presented as a data **structure** in its ...
www.aw.com/cseng/titles/0-201-88795-9/ - 36k - Cached - Similar pages

Courses in Computer Science
... algorithmic analysis; advanced data **structure** design. ... of databases: the entity-**relationship**, network, relational ... design, evaluate, and build user **interfaces**. ...
www.cs.oberlin.edu/coursesmain.html - 16k - Cached - Similar pages

Draft specification for 22C:17
... Abstract **interface** specifications. ... **Relationship** with stack data **structure**: stack of activation ... for important algorithms and data **structures**: binary search ...
www.cs.uiowa.edu/~epley/Draft.22C17.html - 3k - Cached - Similar pages

Computer Science Department
... Advanced data **structures** (eg, balanced trees, **hash** ... eg, support for graphical **interfaces**, concurrency, non ... **Relationship** of language design and implementation. ...
merlin.cs.uah.edu/graduate/msrequire.asp - 12k - Cached - Similar pages

[PPT] Chapter 1: Introduction
File Format: Microsoft Powerpoint 97 - View as HTML
... Schema – the logical **structure** of the database. ... **Relationship** set depositor associates customers with ... Application **program interface** (eg ODBC/JDBC) which allow ...
www.cs.bu.edu/faculty/gacs/ cs112b1/lectures/ondatabases.ppt - Similar pages

Observer Design Pattern
... 0 defines the function Update() that **interfaces** with the ... ConcreteObserver() terminates the **relationship** with the ... However, their class **structure** can still be ...
www.cs.clemson.edu/~malloy/ courses/patterns/observer.html - 17k - Cached - Similar pages

[PDF] Glossary of Object Oriented Terms
File Format: PDF/Adobe Acrobat - View as HTML
... **interface**: The set of all signatures (public methods) defined for ... Is-A: A **relationship** in which the derived class ... linked list: A data **structure** in which each ...
www.owlnet.rice.edu/~mech517/Books/oop14.pdf - Similar pages

MIT/GNU Scheme Reference: Associations
... has a rehash threshold that specifies the **relationship** of the ... provide a lower-level **interface** to the ... or trees from other aggregate **structures** like association ...
www.gnu.org/**software**/mit-scheme/ documentation/scheme_12.html - 101k - Cached - Similar pages

Team Pantheon (09-5), **Software** Design Specification 3.0
... 4.0 involves reuse of code and **relationship** to other ... math libraries, hardware registers, interrupt **structures**, and system ... query to the SQL **Interface** and then ...
www.cs.utexas.edu/users/s2s/latest/icc3/doc/SDS.html - 26k - Cached - Similar pages

h g gec e ch h e e e h ef ce c e f e

Gooooooooooogle ▶

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Free! Get the Google Toolbar. Download Now - About Toolbar

relationship interface structure so Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

©2004 Google

h g g e c e ch h e e e h e f ce c e f e

L Number	Hits	Search Text	DB	Time stamp
-	1075	719/310,312,316.ccls.	USPAT; US-PGPUB; EPO; JPO	2004/08/18 09:23
-	9184	709/201-203,217-218.ccls.	USPAT; US-PGPUB; EPO; JPO	2004/08/16 12:12
-	125	719/310,312,316.ccls. and 709/201-203,217-218.ccls.	USPAT; US-PGPUB; EPO; JPO	2004/08/16 12:12
-	10134	719/310,312,316.ccls. or 709/201-203,217-218.ccls.	USPAT; US-PGPUB; EPO; JPO	2004/08/16 13:36
-	10134	719/310,312,316.ccls. or 709/201-203,217-218.ccls.	USPAT; US-PGPUB; EPO; JPO	2004/08/16 13:36
-	39	(719/310,312,316.ccls. or 709/201-203,217-218.ccls.) and (interface same relationship same (sturcute or hash or table))	USPAT; US-PGPUB; EPO; JPO	2004/08/16 14:42
-	1	("6,263,491").PN.	USPAT	2004/08/16 14:53
-	1	("6,381,628").PN.	USPAT	2004/08/16 14:56
-	1	("6,381,735").PN.	USPAT	2004/08/16 14:58
-	0	COGIN and hunt and microsoft	USPAT	2004/08/16 14:58
-	0	COGIN and hunt and microsoft	USPAT; US-PGPUB; EPO; JPO	2004/08/16 14:58
-	0	DCOM and galen-hunt$.as. and microsoft	USPAT; US-PGPUB; EPO; JPO	2004/08/16 14:59
-	0	DCOM and galen-hunt$.in. and microsoft.as.	USPAT; US-PGPUB; EPO; JPO	2004/08/16 14:59
-	26	hunt$.in. and microsoft.as.	USPAT; US-PGPUB; EPO; JPO	2004/08/16 14:59
-	8	hunt$.in. and microsoft.as. and DCOM and ADPS	USPAT; US-PGPUB; EPO; JPO	2004/08/16 15:00
-	1	("6463508").PN.	USPAT	2004/08/18 09:24

L Number	Hits	Search Text	DB	Time stamp
-	1	hunt-galen.in.	USPAT; EPO; JPO	2003/09/09 17:18
-	12	hunt-galen$.in.	USPAT; EPO; JPO	2003/09/09 17:18
-	16	("3427443" | "3551659" | "4819233" | "5579520" | "5724584" | "5752038" | "5790858" | "5806061" | "5917998" | "5978785" | "5987247" | "6023696" | "6088717" | "6112304" | "6131095" | "6134559").PN.	USPAT	2003/09/09 17:19
-	11	("3427443" | "3551659" | "4819233" | "5021947" | "5579520" | "5748962" | "5752038" | "5790858" | "5978583" | "6011918" | "6044224").PN.	USPAT	2003/09/09 17:25
-	17	("3427443" | "3551659" | "4819233" | "5193180" | "5535329" | "5579520" | "5675805" | "5752038" | "5790858" | "5881268" | "5987247" | "6088717" | "6088732" | "6112304" | "6117188" | "6131095" | "6134559").PN.	USPAT	2003/09/09 17:30
-	12	("3427443" | "3551659" | "4819233" | "5193180" | "5535329" | "5579520" | "5752038" | "5790858" | "5987247" | "6088717" | "6112304" | "6134559").PN.	USPAT	2003/09/09 17:32
-	16	("3427443" | "3551659" | "4819233" | "5193180" | "5247678" | "5535329" | "5579520" | "5675805" | "5752038" | "5790858" | "5978583" | "5987247" | "6044224" | "6088717" | "6112304" | "6134559").PN.	USPAT	2003/09/09 17:33
-	365898	network	USPAT; EPO; JPO	2003/10/02 15:57
-	48787	network and server	USPAT; EPO; JPO	2003/10/02 15:57
-	1185	(network and server) and DLL	USPAT; EPO; JPO	2003/10/02 15:57
-	39	((network and server) and DLL) and IDL	USPAT; EPO; JPO	2003/10/02 15:57
-	36	(((network and server) and DLL) and IDL) and distribut$4	USPAT; EPO; JPO	2003/10/02 15:58
-	34	((((network and server) and DLL) and IDL) and distribut$4) and model	USPAT; EPO; JPO	2003/10/02 15:58
-	34	((((network and server) and DLL) and IDL) and distribut$4) and model	USPAT; EPO; JPO	2003/10/02 15:58
-	23	(((((network and server) and DLL) and IDL) and distribut$4) and model) and meta$5	USPAT; EPO; JPO	2003/10/02 15:59
-	23	((((((network and server) and DLL) and IDL) and distribut$4) and model) and meta$5) and application	USPAT; EPO; JPO	2003/10/02 15:59
-	5	(("5201947") or ("5748962") or ("5978583") or ("6011918") or ("6044224")).PN.	USPAT	2003/10/02 17:02
-	1	("5021947").PN.	USPAT	2003/10/02 17:02

- Galen C. Hunt and Michael L. Scott. The Coign Automatic Distributed Partitioning System. To appear in *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI'99)*. New Orleans, LA, February, 1999.
  Also available as MSR-TR-98-40 in Microsoft Word '97, PostScript, and PDF.

- Galen C. Hunt and Michael L. Scott. A Guided Tour of the Coign Automatic Distributed Partitioning System. *Proceedings of the 2nd International Enterprise Distributed Object Computing Workshop (EDOC '98)*, pages 252-262. San Diego, California, November, 1998.
  Also available as MSR-TR-98-32 in Microsoft Word '97, PostScript, and PDF.

- Galen C. Hunt. Automatic Distributed Partitioning of Component-Based Applications. Ph.D. Dissertation, University of Rochester, Rochester, New York, July 1998.
  Available in PDF.

- Galen C. Hunt and Michael L. Scott. Coign: Efficient Instrumentation for Inter-Component Communication Analysis, URCS Tech Report 648, University of Rochester, Rochester, New York, February 1997.

Printer-Friendly Version Send This Page Add to Favorites Comments

Manage Your Profile | Contact Us

©2004 Microsoft Corporation. All rights reserved. Terms of Use | Privacy Statement

no source code. Determining communication and resource costs for each component is difficult if not impossible.

Even if you can determine an ideal distribution, evolutionary factors may render it obsolete. Future changes might include the topology of computation resources, network latency, or the need to insert new layers in the component architecture. A sound component distribution coded into the application today might become a serious weakness in the future.

Using Coign, you create an application and leave the details of distribution to the system. After aggregating the necessary prefabricated components and writing the necessary connection code, you are ready to create a distribution version of the application. You instruct the Coign binary re-writing tool to insert the Coign run-time instrumentation into your application. You then run your application through a number of profiling scenarios. Coign's instrumentation discovers which components are used in the application by instrumenting the COM libraries and interfaces at run time. The instrumentation measures all communication between application components.

After completing the profiling scenarios, another Coign tool constructs an abstract model of your application's components and inter-component communication (ICC). The abstract ICC model is network independent. You now deliver the application to your customers. When you customers run your application, a small residual piece of the Coign run time embedded in your application distributes the application to minimize communication costs. Communication is minimized by the Coign runtime which automatically tunes the application's distribution to take into consideration currently k latency and available bandwidth.

The effectiveness of a Coign-chosen distribution is based on how closely the profiling scenarios match a user's usage of the application. Because Coign can re-partition and re-distributed applications without access to application binaries, your customers can even re-profiling the application on site to fully tune the application to their needs.

With Coign, you can create a single application and the Coign ADPS automatically adjust its component distribution as network topology and constraints and customer usage patterns change . You aren't locked into coding a single static distribution choice.

Status:

We have completed an initial implementation of Coign on Windows NT. Coign is the first ADPS to automatically distribute binary applications built from components. Coign has successfully distributed three commercial-grade applications: *Microsoft PhotoDraw 2000*, the Octarine word processor, and the Corporate Benefits Sample from the Microsoft Developer Network (MSDN). In all, Coign has successfully distributed more than 2 million lines of component-based applications. In terms of complexity, several of the Octarine scenarios we have tested instantiate over 3,800 components.

From non-distributed applications, Coign has produced distributed applications with communication costs up to 31% lower than other source-free distribution techniques.

Coign has also proven effective in re-optimizing manually distributed applications. In one case, Coign optimized a 3-tier application to reduce communication costs by 54%.

Coign is completely language neutral because it leverages COM's binary standard interfaces and true location transparency. Coign has distributed applications with components written in Visual Basic, C, C++, and x86 Assembler. All of the applications distributed with Coign were created with no *a priori* knowledge of the existence of Coign.

Coign has successfully demonstrated that commercial applications can be automatically distributed without access to application sources. Furthermore, Coign has demonstrated that the internal architecture of existing COM-based applications can be exploited to produce distributed applications with good performance.

Papers:

Microsoft.com Home | Site Map

Microsoft Research

Search: All Research Online Go

Microsoft Research Home
About Microsoft Research
Research Areas
People
Worldwide Labs
University Relations
News
Publications
Downloads
Conferences and Events
Lectures Online
Related Web Sites
Press Resources
Careers
Visiting Microsoft Research
Contact Us

Coign

Executive Summary:

Coign is an automatic distributed partitioning system (ADPS). It converts local COM applications into distributed client-server applications without access to source code. Using scenario-based profiling, Coign discovers the internal structure of an application and cuts the application into client and server components; choosing a distribution that minimizes communication between client and server.

In layman's terms, Coign takes an single-computer application, as shipped on a CD, watches how you use it, then divides it to run simultaneously on two computers to improve its performance.

What is Coign?

Coign is an *Automatically Distributed Partitioning System* (ADPS) for component-based applications. Given either a non-distributed or distributed application built from COM components, Coign will automatically create a new distributed version of the application. Using scenario-based profiling, Coign creates an inter-component communication (ICC) model of the application. Later at execution time, Coign uses the ICC model to partition the application into client and server components and distribute the components across a network to reduce total communication costs. Coign operates on application binaries (.exe and .dll files); analysis, partitioning, and distribution are achieved with no access to application sources. Coign is language independent.

Coign was part of the Millennium Project at Microsoft Research, a project to develop distributed systems that provide a new level of abstraction for application programmers and users, managing machines and network connections for the programmer in the same way that operating systems today manage pages of memory and disk sectors.

From *The American Heritage Dictionary of the English Language*:

coign *noun*

1. A stone serving to form an exterior angle of a wall or other piece of masonry; **a cornerstone**.
2. **A keystone.**

A Typical Scenario:

You are assigned the task of creating a large client/server application. Users access the application from their PCs. The application manipulates data stored on a server. To economize development time, you build the application from a number of prefabricated software components. Using known techniques, you can readily connect the components to create your custom application.

Difficulties arise when you must partition components between host computers. Components can reside on the user's PC, the centralized server, or possibly a departmental compute server. Factors influencing your decision include computation or memory resources available on each host, network bandwidth and latency, and constraints on the location of components with access to sensitive data. Choosing an optimal distribution is further complicated by the fact that the prefabricated components are binary objects with